Full Stack Testing is a Culture: Drive your team to Embrace the Change

Christina Thalayasingam

niro.tina2k@gmail.com

Abstract

Many teams aim only to have a bug free system. Seldom do they believe in delivering true quality. Rarely do they realize that skipping nonfunctional testing like performance and security testing could lead to the breach of your product quality. Many teams still do not understand the importance of Continuous Integration or Continuous testing. This paper will discuss how you could pour the passion into your team to look in all accepts of quality. Move into Full-Stack Testing. Build up a team that values the essence of quality is a culture rather than focusing alone on reporting bugs.

We will discuss ways on making your team walk in the path of Full Stack Testing so that the team knows their vision and the mission. The team should move into a position where any member of the team can understand the testing requirements of the application and execute them. Drive the entire quality team to understand the skills that each of them possesses. There are several skills to assure Quality: Testing, Test Automation, Performance, etc. Generally, there are teams specialized in each of these areas executing each of these test types. Here we need to be able to pick anyone from the quality team to be able to cover any of the above test types.

Full-stack quality engineers are individuals capable of working on all aspects of quality across all the application's layers, using different testing methods. They will think about the many different aspects of product quality, such as functionality, usability, performance, security, and will also be familiar with test automation strategies and technologies. The full-stack QE will have a rich mix of domain knowledge, technical skills, and testing expertise. This is the trend that quality engineering is flowing into today, and we shall discuss on how to get there.

Biography

Christina Thalayasingam has more than 6 years of experience in both functional and non-functional testing. She possesses a development background. Since she has worked on PHP Web Development and Android Mobile Development before taking up Quality Engineering. She has worked in automate testing content management systems for the UK government, point of sales applications, eCommerce application, and clinical trial applications. She was worked on-site in the UK on projects with the UK government sector and major food supply chain management company. Christina is currently working as a Senior Test Engineer at Medidata Solutions by Dassault Systèmes®, an award-winning company that develops and markets software as a service (SaaS) for clinical trials. Also, she has been part of various prestigious conferences, technical meetups and webinars. She is a software testing evangelist.

Introduction

Testing is very important for software development companies. Most of them need to guarantee to their clients that their products are multiplatform, multidevice, and that they will work correctly and efficiently. It is always important to perform all quality checks during the entire development cycle. Conducting all quality checks from a very early stage is essential. Such a requirement demands the quality team to be able to drive, administer, and set all vital quality check. They cannot only limit themselves to the traditional skill sets and techniques to achieve this.

The World Quality Report 2018-19 [1] states that "*Agile, DevOps, automation and artificial intelligence not only require newer skill sets but also make it necessary for Quality Engineers to have multiple technical competencies*". This continuous introduction of new methodologies and technologies makes it a necessity for a corresponding change in the Quality Engineering strategy. To meet the demanded pace and quality requirements of the end-user in today's world quality engineers are transforming to adapt to the industry changes. Digital Transformation (Mobility, Cloud, Big Data, Data Analytics, Customer Experience, etc.) is evolving leads to the need of test engineers to play a major role in checking quality. There is high requirement from test engineers to think of ways to speed processes for timely and swift delivery which leads the use of one the following STLC Automation, in Sprint Automation, Business Process Automation, TDD / BDD, Continuous Testing, etc. Increasing agile and DevOps practices where dynamic collaboration between multiple teams is the key rather than adhering to predefined static quality gate processes which leads the test engineers to acquire capabilities on emerging technologies (AI, ML, Blockchain, RPA, AR, VR, IoT, and others).

The current testing roles are designed to function in silos where manual testers only focus on black-box testing, automation testers are effective only on certain automation tools, non-functional testers only focus on product non-functional without digging deep into product functionality, and so on. To attain the maximum quality the demand of all these skills for a quality engineer has become vital. To adhere to the current pace and to support the transformation and drive organizations to the highest level of maturity QEs require to become Jacks-of-all-trades [2] (who can do many different types of work) in their game, even though a master of some.

This paper narrates how the goal of making a full stack quality engineering team out of a quality engineering team was approached. Make your team believe in full-stack testing. Let them see the benefits that it could bring to them and the way they work. Have the ritual of creating proof of concepts to make your team believe in why it is beneficial. Every transformation to the Full Stack Testing story begins with the realization that the current setup isn't working. This does not only require technical skill set change but a culture change.

Need for full-stack Quality engineers

The fundamental urge for full-stack testing is the adoption of agile product development processes. As software updates are delivered more regularly, Test engineers have less time to complete all the testing. There's not much time towards the end of a sprint, and it's certainly too late to do it months after the feature's development is 'complete'.

Therefore, to achieve fast delivery, testing activities must be blended with the agile process and tasks for the sprint. Said differently, testing is not something that should be done before release; it should be done during every sprint. A story/feature should not be considered 'complete' without testing, and rather than being reactive by looking for bugs after development, full-stack testing will focus on proactively looking for defects throughout the planning, design, and development stages. The Quality engineer needs to think up-front about all aspects of quality and how to test, from the start of the sprint's planning phase. Hence quality engineer must grow their skills to match the new technologies, testing techniques, and methodologies currently in trend. Full-stack testing is focused on defect prevention rather than defect detection.

Tasks for Full-stack Test Engineers

A full-stack test engineer should understand the product well to test it from the user's point of view. Test Engineer should be involved from the early stages of the feature's development so that they can share their advice and experience from the testing and user-perspective. Fewer defects are found after initial development if the test engineers are involved from the very beginning, which includes defining the feature's requirements.

It is a must the test engineer works with the developers to understand a feature's architecture, how it's implemented, and the technologies used. This would help them determine the best ways to test the feature, in all aspects of testing. The full-stack test engineer doesn't need to know the actual code of the application, but a comprehension of the implementation can raise some very useful technical questions. This helps the entire team to think extra carefully when implementing the feature and prevent defects from reaching the end-user.

Full-Stack Testing is required more for defect prevention than defect detection. Making sure that a feature/story has gone through quality check in all layers of the application prevents and non-functional check from new defects occurring even in the future. In this manner defect prevented. Therefore, the yearn for defect detection should be to prevent defects occurring in the future. When underlying issues within application layers are fixed there would be very rare possibility to find issues related to old development in the future. This would help will prevent from wasting time in digging the old skeletons out.

In short, having a Full-stack Test Engineer in your team helps with the following.

- Efficiency – Gives the team the flexibility to do more. Having a single test team that possesses all test skills removes the need for an automation team, a performance testing team, etc. This means there'll be less wait time between testing cycles. This means dedicating work for teams in silos can be avoided, which saves time in knowledge transfers to those teams, etc.
- Efficacy – Full-stack Test Engineer in a team often would have a deep understanding of the product. This means the Test Engineer knows what, where, and how to test. Whereas a traditional test engineer without this knowledge will most likely test a larger portion of the product since they will only have an overview of the product, so doing deep testing in an area might not be within their scope.

Overall, an experienced full-stack QA engineer is also a product expert, quality advisor, and risk analyst.

Build your test team into a Full-stack Quality engineer team

Creating a team of full-stack test engineers requires mastering a lot of skills, which may be challenging for many team members to acquire in a short time. Most of the time a team would not comprise of all members being full-stack test engineers. My suggestion is to start with building a full-stack test team first, where the team has the capabilities to handle any task, such as automation, end-to-end functional testing, performance testing, security testing, and so on, collectively.

Following are the stages of building up a full-stack testing team at the beginning:

1. **Clean the mess in the house -** First of all, clean up the existing mess if any in your current testing process. Process should be created in such a way that a feature should be completely tested in all aspects that would affect the user-experience before is considered to be 'complete'. Precise checklists should be created in order to help test engineers discuss on the types of quality effects around the feature being developed. This would lay the initial foundation stone for the test engineer to create the full-stack test strategy for the feature. It would be advisable to include the software architects and lead developers when create this process, so that the test lead can have a standard list of things to look out for in terms of quality. Aim for goodness, not perfection. Derive test plan templates based on the initial experience and continue to gradually update them as the team learns on how they could do things better.

2. **Start small and aim big -** Do not expect everything to fall in place immediately. Therefore, select very basic features to start with going full-stack testing during the sprint testing. Some feature that would have impact if most aspects, for example, auto-complete on a selection list. The Test Engineer will consider issues such as the time delay to refresh the selection after typing a single letter; how to test the event that triggers a REST call to the backend service; how the UI layer behaves if the backend service is broken; etc. In this way most of the user experience related issues can be covered. Discussing every minute thing of the feature construction would help the test team to test full-stack and deliver it as development 'complete' successfully. Therefore, initial practice on basic features rather than complicated stories can help the test team mature on the process. This will prepare the test team to be able face complicated features.

3. **Weigh out the best fits -** Pick the tools for full-stack testing wisely. Your pick will strongly depend on your requirements, the tech-stack you use in your company and how your daily workflow is handled. If you are thinking of integrating various different type of testing, get a platform that can handle most of it. Choose a framework that works well with the team skill sets and help train them to use it without any issues. The tools you choose is crucial, it can impact many factors like efficiency, adaptability and ability get the maximum out of the benefits of full-stack testing

4. **Have a believing team -** Make your team believe in full-stack testing. Let them see the benefits that it could bring to them and the way they work. Have the ritual of creating proof of concepts to make your team believe in why it is beneficial. Every transformation to the Full-Stack Testing story begins with the realization that the current setup isn't working. Full-stack is key to achieving speed across the pipeline. The commitment is necessary for it to work. Make them believe automation reduces manual errors, and bakes quality into every step of the process.

    a. **Assign good shepherds to lead the way -** Assign leads or senior engineers who can lead the test team in believing and working towards full-stack testing. Let them guide the team, sketch out a meticulous game plan to incorporate the change. Make sure this is followed as a ritual. They should consider the company priority and project allocation when allocating resources and defining the process.

    b. **Full-stack testers are jacks-of-all-trades -** The test team needs to be composed of jacks-of-all-trades in the software testing world. A person who has the basic knowledge in all the important quality checks conducted in an application. A jack-of-all-trade would carry the following pros for the growth of the team:

- **Diverse Skill Set** - Usually, a person with several skills or a jacks-of-all-trades is widely miscommunicated and misinterpreted because of the highly negative impact of the phrase. There's a misconception that people who multitask, are quite likely to fail. However, a very important fact is ignored while criticizing such people, their versatility, and their diverse skill set. Full-stack testing is all about jacks-of-all-trades in the software testing. Their diverse skill sets would be vital for growth of the team.
- **Can Be a Blessing for Initializing Full-Stack** – A test engineer with skill sets in most of the test types could be a great addition to a team that requires to take its move to full-stack testing. This can help the other team mates to work together to understand the process.
- **Adaptability and Flexibility** – They could be added to various scrum teams to help the other teams to work towards full-stack testing their features at the initial stages. Acquiring a number of skills eventually makes them proficient enough to make complete use of every component of their skillset. The expansive knowledge base of such individuals makes them flexible and just as comfortable in doing manual functional testing as they would be load testing, processing test data.
- **Long-Term Learner** - Being a jack of all trades, individuals don't just learn one thing. They go through multiple learning processes of different levels of complexities and durations. This definitely helps them gain the most important skill that any person can have - knowing how to learn. Jack of all trades is often lifelong learners. This can help the need to learn for full-stack testing immensely. Eventually the team will grow.
- **Able to provide basic training to rest of the team** – A full-stack engineer in the team can train the test team with a strong plan to go forward.
- **Great Fit for Leadership** - A leader is always considered to have a lot of experience. A leader who knows almost every aspect of what they do (in this term full-stack testing) will have an edge over someone who rose through ranks doing only one job. They will be best to help show the way to full-stack testing.

c. **Get help from masters of one**
The team needs support from the subject experts since they are going to start treading into new ground in terms of few/many types. They need to get the ideas from the masters of each quality check. If it is possible it is best to recruit someone who has such rare valuable skills, so that she/he could teach/train the team with testing types like maybe performance testing or security testing. Eventually such candidates will grow into becoming full stack testing jack-of-all-trades and create them too.

5. **No more working in silos –** Quality checks do not have to depend on expert silo teams like the API test team or the performance testing team. Now all-important quality checks will be completed by the scrum test team before story/feature is marked as "complete". Therefore, lesser issues will be found when it reaches the silo teams who will be at the end of the SDLC. The team now will be able to exchange roles within them, cover each other, help each other and work together on challenging tasks. This would lead for a more sophisticated full-stack test team.

6. **Continuous learning and scrutiny -** Continuous learning is key to keep improving your full-stack testing practice and culture. Scrutinize from what is new out in the industry and learn and adhere to the approach that helps your team to grow.

7. **Practice what you preach -** Building quality into your SDLC (software development life cycle) requires a commitment to automation to full-stack testing. You know you're on the right track when automation is something you do not just to reduce effort, but to also move faster, and build

quality. As you embrace a culture of automation, it’s bound to transform your testing efforts and result in high-quality apps that are shipped faster.

Full stack to continuous testing

As the team starts practicing full-stack testing on every story after a certain point they will gain expertise in what they do. This is when it is time to think about full-stack test automation. Till this stage it is not necessary for the team to automate all test types. Certain tests should be initially conducted manually or semi-manually till the team gains the in-depth knowledge of it. Once the team has reached a point of expertise and confident, they should look into options to move into the world of automating all regression tests be it functional/non-functional. This would eventually lead the team to continuous testing and that should be the goal from the very start. Continuous testing [2] has immense benefits not only for the test team but to entire scrum/product/project team. Continuously running quality checks will show where the application stands in all aspects of quality.

Benefits of full-stack testing

The full-stack testing team would eventually be a versatile test team who would have gain immense knowledge and experience. This experience will help them trouble-shoot an issue with the full-stack test mind set. They would be working not only on the functionality but also on other quality checks which would be an added benefit where quality checks being slipped or forgotten would not be much of a case anymore. Moreover, having them would help with timely delivery which would be budget-friendly. The full-stack testing team would be able to take complete ownership of the quality of the application.

Setbacks to be mindful of

Full-stack testing has so many benefits and that is what many test teams are currently working towards today. However, there are certain things that we need to be mindful of to make sure that we get the right output. The time of the team should be managed precisely, and no training nor learning should their affect productivity. There could be confusion on roles and responsibilities in the absence of certain team members. Those need to be clarified and communicated clearly. Finally, as new trends flow in it could be difficult to keep up with them. However, it is best to delegate members to monitor trends on each aspect of quality so that the team can be aware of the latest and eventually scrutinized the current practices.

References

1. https://sii-concatel.com/wp-content/uploads/blog_enjoyit/2018/10/World-Quality-Report-2018-19.pdf
2. https://dev.to/katieadamsdev/jack-of-all-trades-or-master-of-one-56b2
3. Continuous Testing - https://www.tricentis.com/products/what-is-continuous-testing/
4. https://huddle.eurostarsoftwaretesting.com/need-for-full-stack-qa-in-2020/
5. https://syscolabs.lk/blog/full-stack-quality-engineering/